An Approach Towards Designing A Prescriptive Analytical Logic Model for Software Application Root Cause Analysis

Hoo Meng Wong
Sagaya Sabestinal Amalathas Tatana Zitkova
hoomeng@hotmail.com

Abstract: Most of the business operations utilize software application heavily in today’s business world. Software application cannot afford to stop its functioning in the production environment during business hours. However, it is hard to restore the software application service without identifying the root cause. Software application error can occur either within the software application layer or any other factor outside the software application layer. With an enterprise level of software application, the complexity is high as it involves multiple tiers and many middleware are running in the environment. The root cause analysis of software application will be easily prolonged due to more than one log file required. The duration on restoring the software application service back to the users can be hard to determine whenever error is occurred. Therefore, the objective is to mitigate prolonging on conducting root cause analysis activity. Indeed to identify the root cause of software application error in a more accurate manner, and shorten the duration of root cause analysis activity conducting on software application error. A Prescriptive Analytical Logic Model incorporates with Analytic Hierarchy Process (AHP) is proposed. This logic model contributes a new knowledge in the area of log file analysis to shorten the total time spent on root cause analysis activity. At the same time it contributes the knowledge in AHP to close the knowledge gap.

Keywords: Application log analysis, Log file analysis, Analytic Hierarchy Process.

1. INTRODUCTION

The computing era evolved since data-processing until today, and software application is still required to host on a server box (either on a physical server box or a virtual server box). Server is required to connect with network communication, it is whether a Local Area Network, LAN, or a Wide Area Network, WAN. At the same time server is required to store its data at the location either in a file or in a database. The database is another software application that focuses on handling and processing data into collection of information under an organized manner for accessing. This database can be hosted on another server box. Due to the complexity of having hardware (physical and virtual hardware), Operating System, Networking for software application communication, and database to allow the software application to store its data. This would make the support team facing much more challenge and difficult to identify the valid software application error and to keep up the software application execution up-time to support the business operation. As per the figure 1.1, Operating System is the middle layer that communicates between software application and server resources. Software application has to interact with Operating System to obtain allocated server resources such as CPU, Memory and Hard Disk Space to handle software application processing, which means software application has highly dependency on server resources. Without the server resources, software application will not execute by itself. By depending software application log file alone to conduct root cause analysis would definitely be insufficient whenever error occurred beyond the layer of software application. Hence a proposed research for a new model is required.

Figure 1.0 - Software Application is required sufficient server resources to execute all its functionality.

2. Background of the Study

When a business company adopts Information Technology (IT) as a tool to enable its business, many required software applications would be purchased into the organization to sustain the daily business operation. Software applications cannot afford to have downtime as it can cause the business operation to cease. This has been stated clearly at Labels: Data Center, Downtime, www.evolven.com (2014). Business company can have an option to own its support team that provides software application support to whichever business-as-usual (BAU) system running in the organization. On another option is that business company can adopt the same software application support service from a service provider which is having a dedicated support team running in different shift timing. However, whichever option is chosen, the time spent on analyzing the log file and identifying the root cause of it will be time consuming. This is because if the analysis activity gets prolonged before the decided resolution steps can be applied to fix the software application error. Users of the software application are the party who suffer. Eventually it impacts the entire business operation. To support this
problem statement, there are comments obtained from the Internet search regarding to the time consuming on application trouble-shooting, which are REDDIT (2015) and StackOverflow (2015). Note that the total time taken to resolve the software application error consists of the duration for conducting the root cause analysis activity and the duration for applying the resolution. This approach may or may not involve the Software Development Life Cycle (SDLC) activity for developing the fix by depending on the requirements of the fix. Hence, the proposed resolution is focusing on shortening the time consumption duration on conducting the root cause analysis activity.

3. Scope of the Study
To the best of my knowledge, whenever a software application is executing, it logs events including error event into its log file. However anything that happens beyond the software application’s boundary, the software application may not be possible to log that kind of events into its log file. The supporting reason can be in a situation that server is lack of server resources such as CPU, memory, hard disk space, or break down in network communication and etc. The important point to stress out here is that if the root cause analysis activity is solely focusing on software application log file alone as the input information, it is insufficient. Under such circumstance, multiple logs or input information from different software applications are required. The input information can be:

- Performance and Capacity monitoring system,
- Server and Networking monitoring system, and
- Configuration Management system.

By obtaining all the required log files from various software applications, this activity must not create any disturbance to the involved software applications during activity execution. Hence the exact amount of logs from various software applications or databases must be exacted and integrated to form a dataset for the input information of the root cause analysis. Indeed, the objective of this proposed research is to shorten the duration of conducting root cause analysis, and the sub aim is to identify the related log files as input data of the root cause analysis accurately. Finally the output is the Prescriptive Analytical Logic Model. The objective is set based on a problem, which is the longer duration for conducting root cause analysis, and how it can be quantified to determine that it is a real problem. Therefore, a simple mathematics approach can be used to evaluate the statement of problem.

4. Significant of the Study
The proposed design of the logic model basically is looking for the contribution and the benefit that can be gained from the research. As for the contribution to the business, by lower down the risk of the software application error, and improve the reliability of utilizing the software application would bring business advantages to compete in today’s business industry. This is because with today’s rapid business competitive world, time consuming on analysis and trouble-shooting activities is unacceptable. This is a continuous battle for the support team to face day-to-day software application error challenge in order to provide reliable up-time for the software application utilized in the business organization. On the other hand, business companies can still continue to utilize their existing software applications (without incur any additional operation budget) and at the same time to allow the companies to save the investment budget on spending the capital amount to replace all or partial of the software applications and re-training their users on using the new software applications. This propose model not only can bring the above benefits to business industries but other industries which are using software application for their daily operation, they need to have the software application error fixed without further re-occurrence. Furthermore, it is also good to look at the perspective of the technical beneficial. Over the years there were various researches had been done at this area such as consolidate the logs or integrate the logs for analysis but there had been very attempts to propose a model to deliver a complete package for analyzing and fixing software application error which consists of the activities such as log integration, error analysis, decision making of preferred resolution, and automated on applying the error fix. There is a great potential in this research which brings contribution to business intelligent studies.

5. Statement of the Problem
To conduct the root cause analysis activity on software application error, it is time consuming to identify the valid error.

6. Research Objectives
The primary objective is to mitigate prolonging on conducting root cause analysis activity. There are also different perspectives along with the research objective. There are shown as follows:-

**Business Perspective**

i. To reduce the duration of root cause analysis activity and lead to shorten the total time taken duration of software application downtime.

ii. To mitigate the same software application error for re-occurrence.

iii. To shorten the total software application downtime percentage in a financial year.

**Technical Perspective**

i. To shorten the unnecessary time consuming on conducting root cause analysis.

ii. To improve accuracy to identify the software application error’s root cause.

iii. To identify the valid software application error accurately by applying Analytic Hierarchy Process (AHP) for decision making process.

iv. To decide the best resolution to the software application error by applying Analytic Hierarchy Process (AHP) for decision making process.

v. To improve root cause analysis experience with the input of past analyzed result and resolution activities stored in the knowledge-based database.

**Knowledge Contribution Perspective**

i. To deliver a new Prescriptive Analytical Logic Model incorporates AHP in it. This proposed model consists of the proposed algorithm, and the proposed algorithm will facilitate development of multiple modules by using programming or
scripting language. Which means each module will be carried out specific set of functions, and all these modules will be utilized to form a complete root cause analysis activity towards to achieve the outcome which is to shorten the analysis time spent duration.

7. Literature Review

Stewart, D (2012) is focusing on debugging real-time software application error using logic analyzer debug macros. Eick, S, Nelson, M, and Schmidt, J (1994) they are focusing on presenting the error logs in a readable manner. Wendy, P and Dolores, R (1993) suggested to focus on error detection in software application at the time of software development and maintenance. Salfner, F and Tschirpke, S (2015) are focusing on analyzing error logs by applying the proposed algorithms in order to predict future failure. Murinov, J (2015) had attempted to integrate multiple log files from various software monitoring tool and network devices for better root cause analysis on Web application error, however there is no proposed model stated in his research. Only Valdman (2001) had showed more detailed feature whereas other studies indicated intentions or suggestions. By reviewing the mentioned approaches and suggestions on conducting root cause analysis activity on software application error, mainly it is either focus on software application error log file alone for root cause analysis activity, or suggested the software application error analysis activity should be built in during the software development process. Indeed, these approaches and suggestions are narrow if the software application root cause is beyond the software application layer. On the other hand, based on the comparison of market log file monitoring product such as Solarwinds Log Analyzer, Logstash and Cisco AppDynamics that they claim to have predictive analytic ability, and create a graphical visualization for trending report. However, any error beyond the common errors, the decision making to identify the valid error is still on human decision. On the AHP side, Vaidya and Kumar (2004) had reviewed Analytic Hierarchy Process (AHP) with 150 application papers in total, and there is no knowledge in applying AHP on identifying valid software application error. Since there is no published article showing that AHP had been applied to software application error analysis, the proposed research is to fill up the knowledge gap. AHP was developed by Thomas L. Saaty stated in Wikipedia (2015), with the proposed algorithm of the prescriptive analytical logic, the proposed algorithm can utilize and apply the theory of AHP to shortlist the best resolution, and to be proposed or to be applied against the software application incident automatically or manually. By referring to R. W. Saaty (1987), the proposed algorithm can apply the three principles in identifying valid software application error, and on the other hand in identifying preferred resolution. The three principles which are the decomposition principle, the comparative judgments, and the synthesizing priorities. In addition, Vaidya and Kumar (2006) had provided the discussion on how to apply AHP under the analytic hierarchy process, and this would help us to understand how the proposed algorithm would apply it in the scenario after identifying the valid software application error and identifying preferred resolution during the analysis process under the proposed model behind the prescriptive analytical logic. There are published articles and journals available for the analysis and studies at the area of AHP, which is good to investigate how the past suggestions and techniques of AHP can be adopted and utilized similarly in this proposed research activity.

<table>
<thead>
<tr>
<th>No.</th>
<th>Title</th>
<th>Author</th>
<th>Analysis and Discussion</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Analytic Hierarchy Process: An overview of applications</td>
<td>Omkarprasad S. Vaidya and Sushil Kumar, 2004.</td>
<td>This is a review paper for the applications of Analytic Hierarchy Process (AHP) with 150 application papers in total are referenced and reviewed. Carefully went through the detail of the review paper and cross checked the references of this paper, there is no published paper discussing about Log File Analysis using AHP approach for decision making process.</td>
</tr>
<tr>
<td>2</td>
<td>A Spatial Analytic Hierarchy Process for Identification of Water Pollution with GIS Software in an Eco-Economy Environment</td>
<td>Razvan Serbu, Bogdan Marza and Sorin Borza, 2016.</td>
<td>This paper used AHP multi-criteria method to evaluate each sampling station’s physicochemical parameters, and based on the obtained result to place respective river in the water quality classes. The research was using GIS software which was allowed an integrated automatic data collection system and displays interactive results.</td>
</tr>
<tr>
<td>3</td>
<td>Applying the Analytic Hierarchy Process (AHP) to build a strategic framework for technology roadmapping</td>
<td>Nathasit Gerdsri and Pundar F. Kocaoglu, 2007.</td>
<td>This paper introduces the Technology Development Envelope (TDE) with the AHP as part of the proposed framework that structured by obtaining strategic information on the development of technologies. Then, this strategic information can be utilized to evaluate the value of each technology based on the impacts of its characteristics on an organization’s objectives.</td>
</tr>
<tr>
<td>4</td>
<td>Applications of the extent analysis method on fuzzy AHP</td>
<td>Da-Yong Chang, 1996.</td>
<td>This paper introduces the use of triangular fuzzy numbers for pairwise comparison scale of fuzzy AHP as the new approach.</td>
</tr>
<tr>
<td>5</td>
<td>Decision making with the analytic hierarchy process</td>
<td>Thomas L. Saaty, 2008.</td>
<td>This paper introduces and explains AHP as a theory of measurement through pairwise comparisons and relies on the judgements of experts to derive priority scales.</td>
</tr>
<tr>
<td>6</td>
<td>The Analytic Hierarchy Process What It Is And How It Is Used</td>
<td>R. W. Saaty, 1987.</td>
<td>This paper introduces the AHP as a measurement on ratio scales, and discuss some of the ideas relating to AHP process and ramifications.</td>
</tr>
<tr>
<td>7</td>
<td>The Analytic Hierarchy Process:</td>
<td>Dr. Benito L. Tehankee and De La Salle,</td>
<td>This paper advocates the teaching of AHP to the business and management students, and stressing AHP can be incorporated multiple quantitative and qualitative criteria in</td>
</tr>
</tbody>
</table>
Capturing Quantitative and Qualitative Criteria for Balanced Decision-Making

University. 2009.

This paper is from the engineering application perspective to examine some of the practical and computational issues encountered when the original AHP method is applied. In addition, the paper also discussed that using Revised-AHP (was introduced by Belton and Gear (1983)) to overcome the deficiency of the original AHP when the original AHP was used in the pairwise comparisons. Further more, on the Multi-criteria decision-making (MCDM) problem whenever alternatives appear to be very close with each other, this paper suggested to use Revised-AHP as it would be the most efficient method to examine in such situation.

Using the Analytic Hierarchy Process for Decision Making in Engineering Applications: Some Challenges


After investigating the area of knowledge in AHP based on the publications, so far there is no knowledge in applying AHP on identifying valid software application error. This has clearly shown a strong support reason to conduct the proposed research in this area in order to fill up the knowledge gap, which is applying AHP on software application error analysis to decide the valid software application error. Hence, with the Prescriptive Analytical Logic Model, it carries high potential on knowledge contribution as the new logic model has the algorithm to be able to identify the root cause of the software application error more accurately under the AHP processing approach, and deliver the outcome to shorten the analysis duration of conducting root cause analysis activity during the software application downtime.

8. Proposed Design

The research is to propose a Prescriptive Analytical Logic Model design that consists of a set of proposed algorithm. It is used to analyze the root cause of the software application from the log events retrieved from various software application log files. With this proposed logic model design, it is aiming to resolve the statement of problem.

The software application error can be categorized under the predefined categories of possible errors depending on the error aspect. The objective is to focus within known category of error and narrow down the root cause. There are changes that error is hard to determine and hard to match into any known error category. Under this circumstance it is good to consider a so called “Other” category to categorize the unknown error into it. On the next analysis activity, the design of logic model is required to look for any past analyzed result for the same issue from the knowledge base database. Base on the current information is obtained, that it is the error matched out from the error category, and the past

Table 1.0 – Analysis and discussion of published articles related to AHP approach.

<table>
<thead>
<tr>
<th>Activity</th>
<th>Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>By given the permission, retrieve the log events from the involved software application log files.</td>
</tr>
<tr>
<td>2</td>
<td>Identify whether the newly reported software application error is first time occurrence or re-occurrence by cross-checking the database which is associated to the prescriptive analytical logic.</td>
</tr>
<tr>
<td>3</td>
<td>Identify possible log data and select the necessary log data for analysis under the defined software application error classification.</td>
</tr>
<tr>
<td>4</td>
<td>Allocate weight to each possible software application error based on Analytic Hierarchy Process (AHP).</td>
</tr>
<tr>
<td>5</td>
<td>Shortlist the software application error under the highest weight.</td>
</tr>
<tr>
<td>6</td>
<td>Analyze the selected log data for shortlisted software application errors and define possible resolution option.</td>
</tr>
<tr>
<td>7</td>
<td>Allocate weight to each possible resolution option based on AHP.</td>
</tr>
<tr>
<td>8</td>
<td>Shortlist the preferred resolution option under the highest weight.</td>
</tr>
<tr>
<td>9</td>
<td>Deploy the preferred resolution option to fix the software application error under the predefined condition.</td>
</tr>
<tr>
<td>10</td>
<td>Store the analysis result and resolution action into a database which is associated to the prescriptive analytical logic for future reference and knowledge base activities.</td>
</tr>
</tbody>
</table>

Table 2.0 - Proposed process activity under proposed algorithm.
analyzed result for the same issue retrieved from the knowledge base database. The analysis process will identify the need of additional required information retrieved from various software application databases. These software application databases are Configuration Database, Production Support Ticketing System Database, and Application and System Monitoring Database, and etc). With all the obtained information, the analysis process must identify the possible involved areas with this error. Whether it is solely being fixed at the software application layer, or other layers such as:-
- Hardware capacity layer (i.e. CPU, memory, hard disk),
- Operating System layer,
- Network layer,
- Database layer, or
- Any combination which more than one layer.

Again, the analysis process will identify the need of additional required information retrieved from specific or various log files from different layers. There is also circumstance that when a major software application error occurs, it would also rigger a list of related and even non-related errors or warning messages in the software application log, with such the analysis process will need to retrieve software application log information under a specific time duration from the software application log file as for the input information of the analysis process. The AHP process is applied to allocate weight to the software application error as well as errors found from different layers under the same “timing”. Therefore, AHP process is playing an important role to allocate weight to each possible error based on the error aspect. Whenever the weight is allocated on each error, the AHP process will shortlist the error under the highest weight. Once the shortlisted error is determined, the AHP process will proceed to evaluate all the possible resolutions based on the weight allocation to each possible resolution. Finally, the AHP will shortlist the resolution on the highest weight. If the possible resolution is more than one, then the PAL will generate a report by providing the possible resolution option. The PAL is designed to have two configuration files, one is designed to record down the common errors and room to cater for new found error. Then, another one is designed to record down the resolution steps for the common errors as well as room to cater for human to provide new resolution steps based on the new found error. These preferred resolution actions under the resolution are manual, semi-automatic, and fully automatic. The PAL must obey the configurations in order to carry out the resolution under the preferred actions. Especially for the “Complex Analysis”, it is required to store the analysis result and the resolution action into a knowledge based database which is associated to the PAL for future reference. Regardless the analysis activity is under simple analysis or complex analysis, the knowledge based is catered for upcoming analysis activity and fine tuning the logic. It is mainly for improving the decision making on selecting the final resolution to be applied to the new found error in the near future. Lastly, from a training purpose perspective, since the knowledge base of the proposed PAL will play an important role for the entire analysis process during the error analysis activity. It can be utilized as training material to train any new joining IT support engineers to pick up the support experience rapidly. At the same time it benefits for those who are experience support engineer to improve the knowledge of software application support. To ease of retrieving the information from knowledge base, the knowledge base database can be either stored locally at the database server or in the cloud. This is because to access the knowledge based in the cloud, the information is accessible at any time.

9. Prerequisites of the Proposed Model

- Multiple Tiers Environment
  In this proposed research, it is required to run on a multiple-tier environment which involves Web tier, application tier, and database tier. The proposed algorithm will be implemented into a software plug-in component sitting at the logic tier to integrate all the required log information from various software applications and store in a separate location for information retrieval later. This will help on retrieving log event from log file as input data from various software applications when the root cause analysis activity is triggered at the logic tier.

- NTP Server
  Whenever a software application involves in a multiple-tier environment, it is good to standardize the server time across all the involved servers. The supporting reason is that the proposed Prescriptive Analytical Logic Model is required to collect log data from different software applications based on the specific given time as the primary key, and therefore the server time on all the involved servers are required to have the network time synchronization. As per Masterclock (2016), “Accurate time stamping is key to root-cause analysis, determining when problems occurred and finding correlations. If network devices are out of sync by a few milliseconds or, in extreme cases a few seconds, it can be very difficult for network administrators to determine the sequence of events.”.

- Technologies of the Proposed Model
  The proposed prototype of the software plug-in component will be coded using Java Programming as this programming language is platform independent. At the platform, it required to install Java Runtime Environment (JRE) and configure the Java home path and memory allocation for the JRE.

10. Conclusion

Today most of the software applications that come along with the event logging feature. This feature is mostly embedded into the software application during the software development cycle. However, in the scenario whenever server resources are running low, software application will face issue during its execution. Other issues can even come from network connectivity, no response from database server and etc, which the issues are beyond the software application layer. Hence, if the root cause analysis is solely depending on software application error log file alone, it is definitely insufficient and the result cannot be accurate. By improving the software root cause analysis technique into a more human-like intelligent, a proposed logic model is required. The objective of this proposed research is to reduce the
duration of conducting root cause analysis, and the sub aim is to identify the related log files as input data of the root cause analysis accurately. Finally the output is the Prescriptive Analytical Logic Model (PAL). The objective is sat based on a problem, which is the longer duration for conducting root cause analysis, and how it can be quantified to determine that it is a real problem. Therefore, by using a simple mathematics approach, it helps to evaluate the statement of problem.

References


[95] Superuser. 2018. Is it possible to access the Windows event log if the system is unbootable? https://superuser.com/questions/76340/is-it-possible-to-access-the-windows-event-log-if-the-system-is-unbootable (Accessed on 9th April 2018)


